

BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/0 17/2  
COMBINED QUARTERLY TECHNICAL REPORT NUMBER 25. SATNET DEVELOPME--ETC(U)  
MAY 82 J F HAVERTY MDA903-80-C-0353  
BSN-5003 ML

**UNCLASSIFIED**

July 1978

END  
DATE  
FILMED  
7-8  
DTIC

**Bolt Beranek and Newman Inc.**



**Report No. 5003**

**Combined Quarterly Technical Report No. 25**

**SATNET Development and Operation  
Pluribus Satellite IMP Development  
Remote Site Maintenance  
Internet Operations and Maintenance  
Mobile Access Terminal Network  
TCP for the HP3000  
TCP for VAX-UNIX**

**May 1982**

**Prepared for:  
Defense Advanced Research Projects Agency**

**DTIC  
ELECTE  
JUN 18 1982  
S D  
H**

**DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited**

**82 06 13 014**

**AD A115773**

**DTIC FILE COPY**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A115 773	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  COMBINED QUARTERLY TECHNICAL REPORT No. 25	5. TYPE OF REPORT & PERIOD COVERED Quarterly Technical 2/1/82 to 4/30/82	
7. AUTHOR(s)  J. F. Haverty	6. PERFORMING ORG. REPORT NUMBER 5003	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238	8. CONTRACT OR GRANT NUMBER(s) MDA903-80-C-0353 & 0214 N00039-82-C-0412 N00039-80-C-0664 N00039-81-C-0408	
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS  ARPA Order No. 3214	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DSSW Room 1D, The Pentagon Washington, DC 20310	12. REPORT DATE May 1982	
NAVELEX Washington, DC 20360	13. NUMBER OF PAGES 71	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE/DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Computer networks, packets, packet broadcast, satellite communication, gateways, Transmission Control Protocol, UNIX, Pluribus Satellite IMP, Remote Site Module, Remote Site Maintenance, shipboard communications, VAX, ARPANET, Internet.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  This Quarterly Technical Report describes work on the development of and experimentation with packet broadcast by satellite; on development of Pluribus Satellite IMPs; on a study of the technology of Remote Site Maintenance; on Internetwork monitoring; on shipboard satellite communications; and on the development of Transmission Control Protocols for the HP3000 and VAX-UNIX.		

DD FORM 1 JAN 73 1473

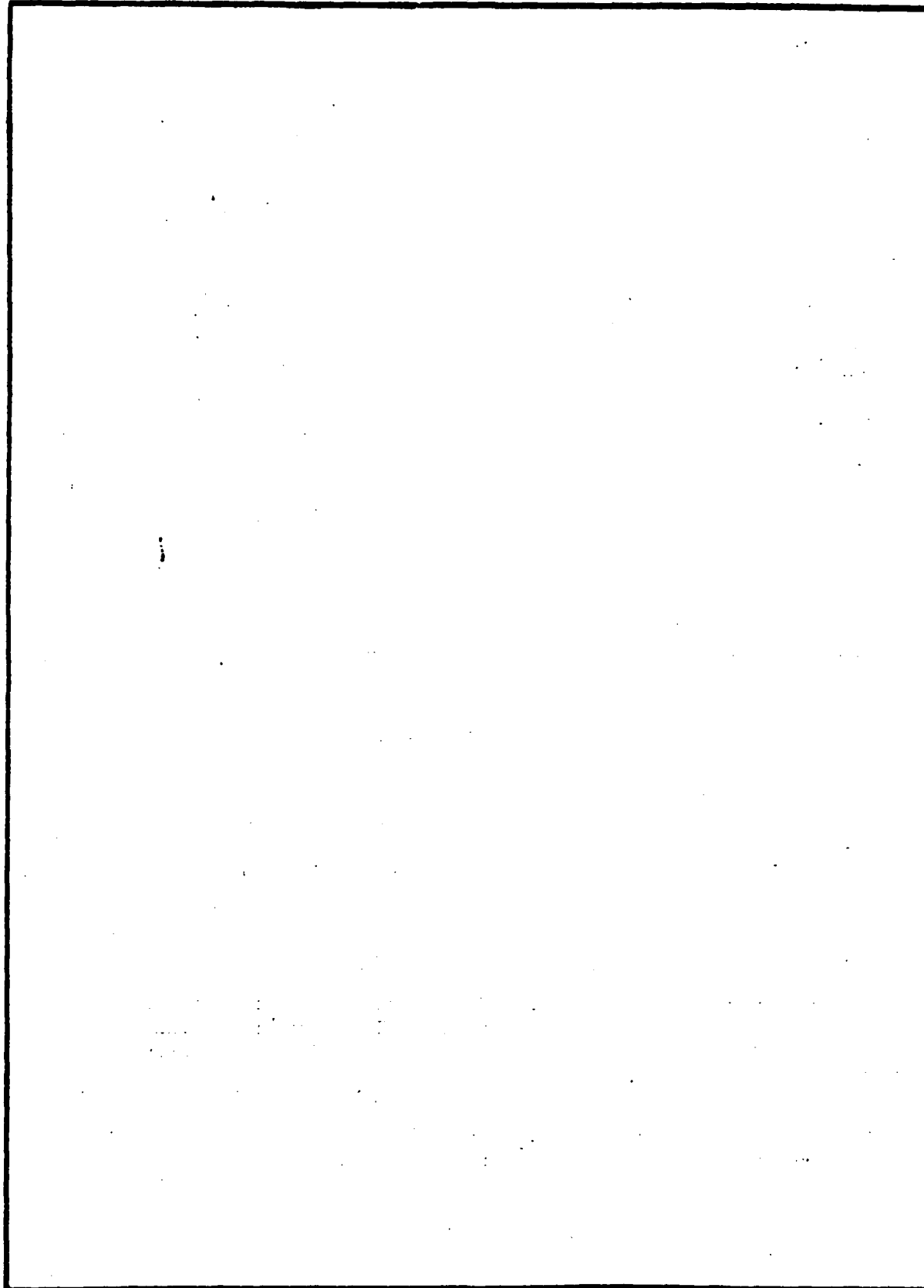
EDITION OF 1 NOV 68 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)**



**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)**

Report No. 5003

COMBINED QUARTERLY TECHNICAL REPORT NO. 25

SATNET DEVELOPMENT AND OPERATION  
PLURIBUS SATELLITE IMP DEVELOPMENT  
REMOTE SITE MAINTENANCE  
INTERNET OPERATIONS AND MAINTENANCE  
MOBILE ACCESS TERMINAL NETWORK  
TCP FOR THE HP3000  
TCP FOR VAX-UNIX

May 1982

This research was supported by the Defense Advanced Research Projects Agency under the following contracts:

N00039-82-C-0412  
MDA903-80-C-0353, ARPA Order No. 3214  
MDA903-80-C-0214, ARPA Order No. 3214  
N00039-80-C-0664  
N00039-81-C-0408

Submitted to:

Director  
Defense Advanced Research Projects Agency  
1400 Wilson Boulevard  
Arlington, VA 22209

Attention: Program Management

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

## Table of Contents

1	INTRODUCTION.....	1
2	SATNET DEVELOPMENT AND OPERATION.....	2
2.1	C/30 Satellite IMP Microcode Debugging.....	2
2.2	Software Maintenance Operations.....	5
2.3	Hardware Maintenance Operations.....	10
3	PLURIBUS SATELLITE IMP DEVELOPMENT.....	16
3.1	PSAT Stream Synchronization Algorithm.....	23
3.1.1	Background Material.....	23
3.1.2	Design Approaches.....	26
3.1.2.1	The Active Approach.....	27
3.1.2.2	The Passive Approach.....	27
3.1.2.3	Comparison of Approaches.....	28
3.1.3	Fundamental Operation.....	30
3.1.3.1	Stream Database Acquisition Phase.....	30
3.1.3.2	Stream Database Build Phase.....	32
3.1.3.3	Stream Database Recovery Phase.....	33
3.1.3.4	Deferred Setup Processing Phase.....	35
3.1.3.5	Stream Database Clean-up Phase.....	36
3.2	PSAT Host Support Software.....	37
3.2.1	Old PSAT Host Software.....	38
3.2.2	New PSAT Host Software.....	39
4	REMOTE SITE MAINTENANCE.....	47
4.1	RSM Accounting.....	47
5	INTERNET DEVELOPMENT.....	51
5.1	Macro-11 Gateway Installations.....	51
5.2	New Gateway Features.....	52
5.3	Packet Radio Gateway.....	54
5.4	UCL Gateway and UCL TAC.....	55
5.5	UWISC Gateway.....	57
5.6	NU Monitoring Work.....	57
5.7	VAN Gateway and Telenet Line.....	59
5.8	Stub Gateway Protocol.....	60
5.9	Internet Meeting.....	60
6	MOBILE ACCESS TERMINAL NETWORK.....	61
7	TCP FOR THE HP3000.....	62
8	TCP FOR VAX UNIX.....	63
8.1	Software Distribution.....	63
8.2	TCP/IP Work.....	64
8.2.1	Debugging.....	64
8.2.2	Enhancements.....	65
8.3	Higher Level Protocol Support.....	68

TABLES

Table 1.....	45
Table 2.....	46

## 1 INTRODUCTION

This Quarterly Technical Report is the current edition in a series of reports which describe the work being performed at BBN in fulfillment of several ARPA work statements. This QTR covers work on several ARPA-sponsored projects including (1) development and operation of the SATNET satellite network; (2) development of the Pluribus Satellite IMP; (3) Remote Site Maintenance activities; (4) Internet Operations, Maintenance, and Development; (5) development of the Mobile Access Terminal Network; (6) TCP for the HP3000; and (7) TCP for the VAX-UNIX. This work is described in this single Quarterly Technical Report with the permission of the Defense Advanced Research Projects Agency. Some of this work is a continuation of efforts previously reported on under contracts DAHC15-69-C-0179, F08606-73-C-0027, F08606-75-C-0032, MDA903-76-C-0214, MDA903-76-C-0252, N00039-79-C-0386, and N00039-78-C-0405.



## 2 SATNET DEVELOPMENT AND OPERATION

As part of our participation in the Atlantic Packet Satellite Experiment (SATNET) during the last quarter, we finished the debugging of the BBN C/30 Satellite IMP microcode, including the modifications for compatibility with the redesigned CMM (Command Monitoring Module) interface of the second generation PSP (Packet Satellite Project) terminal. These operations, the overall SATNET software maintenance operations, and the overall SATNET hardware maintenance operations are described in the following sections.

### 2.1 C/30 Satellite IMP Microcode Debugging

During initial backroom testing of a two-node network configuration, we were unable to establish data transfers between the BBN C/30 and the Honeywell 316 versions of the Satellite IMP, although each machine could hear its own transmissions perfectly well. In the test setup, the Satellite IMPs were interconnected through the satellite channel simulator clocking data at 128 Kb/s. ARPANET connectivity was provided for the C/30 via the VDH port designated host 2 on ARPANET IMP 82 and for the Honeywell 316 via TRAN modems connected to the Digital Equipment Corporation PDP 11/44 testbed gateway. Thus, both sites were able to report to RECORDER on BBNC independent of satellite

channel operations. Having the gateway testbed participating with the SATNET testbed facilitated more realistic test situations.

A Biomation Logic Analyzer was connected between the C/30 Satellite IMP and the satellite channel simulator, using a specially built jig on which test points for the interface signals were mounted. The sampled signals included Transmit and Receive Clocks, Transmit and Receive Data, and GOSIG. When clocking the logic analyzer at a 2-microsecond rate, all transitions of the clock signals could be readily discriminated, while an entire Hello packet would fit conveniently within a sampling screenful.

We quickly determined that both the C/30 and the Honeywell 316 were transmitting the packet framing sequences SYN-SYN-DLE-STX and DLE-ETX with the correct parity--even instead of odd; however, the data signals sent by the C/30 were inverted relative to those of the Honeywell 316. This was traced to the C/30 MMSI daughter board mounted above the C/30 MII universal I/O board having been strapped incorrectly. When the original daughter board was replaced with a correctly strapped daughter board, things improved somewhat; namely, the Honeywell 316 received both its own packets and those sent by the C/30, although the C/30 still missed all packets from the Honeywell 316. Since no

checksum errors were indicated, we inferred that the problem originated in the packet discrimination logic.

Another problem seen was that the C/30 GOSIG signal was asserted beyond the end of the packet. Such a situation adversely affects the procedure for prefixing the packet-start-framing-sequence, where queueing of the first SYN of the following packet occurs during completion of a packet output transfer. We regularly observed that the C/30 transmitted only a single recognizable SYN character at the beginning of a packet.

Two serious bugs were also found in the C/30 microcode. The first bug caused the C/30 to assume that the packet count specified in the message header is the precise number of words in the transfer. In actuality, the first two words in the header and the packet framing and checksum sequences are not included in the count. The second bug impacted the latency requirements of the modem input process; namely, the Satellite IMP would miss an incoming packet if it had not issued a modem input instruction by the time that the packet-start-framing-sequence began arriving. When this bug was eliminated, correct operation of the Satellite IMP required the issue of a modem input instruction by the time that the first header word of an incoming packet began arriving.

After the microcode was modified to fix the problems described above, the C/30 was still missing packets regularly.

When further debugging did not reveal the cause of the missing packets, we removed an extremely complicated section of unused microcode specifically oriented toward MATNET, and only then did the microcode begin working properly.

We also converted the CMM I/O handler in the C/30 Satellite IMP microcode to 8-bit standard RS-232 asynchronous data transfers, a modification required for compatibility with the redesigned CMM interface of the second generation PSP terminal. With this design, a standard data terminal may be used to check data transfers either to the PSP terminal or to the Satellite IMP. Since both types of PSP terminal equipment are expected to be in the field for some time, a software switch in the microcode selects whether the CMM I/O handler supports the old style or the new style data transfers. Debugging of the new microcode was performed with COMSAT personnel on the Clarksburg C/30 Satellite IMP.

## 2.2 Software Maintenance Operations

We finished the conversion of the Satellite IMP source software to a form compatible with a C/30 cross assembler, currently residing in BBN C/70 UNIX\* machines. This conversion

---

\*UNIX is a trademark of Bell Laboratories.

required rewriting many of the assembler macros and relocating many of the software modules to different pages in memory. Contained among the new assembler macros are those which more directly correspond to sequence operations needed in the new more powerful and more flexible I/O architecture, designated the Native Mode Firmware System (NMFS). Hence, implementation of NMFS, scheduled prior to implementation of 128 Kb/s operation, will be facilitated.

As a preliminary test of the software conversion procedure to the new assembler, the SATNET VDH Loader/Dumper was converted first and successfully released to Etam. Subsequently, after the normal hours allocated daily to the support of ARPANET traffic from European users, testing of the new Satellite IMP software was conducted in the field. The last day of testing was so successful that we left the software in all field sites and declared to the community the release of Satellite IMP versions 3.5:0 for Honeywell 316s and 4.5:0 for BBN C/30s.

Unfortunately, a problem surfaced between the Goonhilly Satellite IMP and the UCL gateway, requiring the release to be withdrawn. The problem was later traced to the Host-SATNET protocol accept/reject messages not being handled properly. Since the MACRO-11 gateway installed at BBN was more or less fault tolerant of this particular problem, it was hidden to us in

all of our tests at Etam. Only when the software was released to Goonhilly did we discover that the BCPL gateway installed at UCL was not tolerant of this problem at all.

Later, after the above problem was fixed and the new software had been released to the field for several days, we discovered two more problems, requiring us to withdraw the release again. First, because a range check was incorrect for determining whether a pending reservation is within the correct buffer pool, both Etam and Tanum crashed several times. Second, because the Satellite IMP did not protect itself against addresses outside its range of acceptable addresses, Etam crashed as a consequence of a malformed packet being accepted having proper format but an illegal address. Both problems were patched, and the new software was permanently released to the field.

Until now, memory constraints in the Honeywell 316 limited Satellite IMP support to only two hosts, one of which was a Host-SATNET Protocol real gateway host and the other an internal gateway module. In anticipation of connecting several gateways to a single C/30 Satellite IMP, the Host-SATNET Protocol Module has been modified to accommodate the interfacing of more than one real host; when needed, interconnection requires only the insertion of appropriate information into host tables. As

before, physical interfaces used by the hosts can be altered at any time by modifying the modem function tables.

The front panel alternate lights display of the three-host Satellite IMP has been modified to indicate the status of the third host. Since by convention the leftmost bits are reserved to show the up/down status of all hosts and since the internal gateway is the last of the hosts, its position in the lights display moves, depending on how many other hosts are defined. In particular, Satellite IMP version 4.5:0 represents the internal gateway by bit #13 (third from the left) rather than #14 (second from the left).

Because the TOPS-20 RECORDER program was not designed to monitor more than two hosts on a Satellite IMP, only two hosts can currently report monitoring information to RECORDER, although the host selection may be altered at any time via a table in the Satellite IMP. When NU monitoring becomes operational, all three hosts will be monitored simultaneously.

Among the other changes made in creating Satellite IMP versions 3.5:0 and 4.5:0 are software modifications, bug fixes, and incorporation of patches developed since the last software release. These changes are described below.

- o The TTY handler has been modified to default to the local DDT and not to the isolated Satellite IMP loopback test. Switching between the local DDT and the isolated Satellite

IMP loopback test has been changed to typing the commands D and I, respectively, on the TTY.

- o A software bug, which potentially created a race condition, was eliminated. Specifically, a message was deleted too early forcing the Satellite IMP to crash if reception of the Host-SATNET ACK preceded the VDH ACK in the Host-SATNET Protocol Module.
- o Discrepancies between a message length and the specified length in its associated reservation will cause a trap rather than a crash.
- o Traps have been inserted in the Host-SATNET Protocol Module for the messages: (a) host up; (b) host down; and (c) host reset.
- o Traps reporting normal activity in the SATNET Service Host have outlived their usefulness and have been deleted.

A new version of the RECORDER program was released, in which changes in the stream count are no longer reported as "events". Extensive use of the facility for providing automatic stream service for gateway traffic has created greater stream activity, such that the number of streams in use is no longer a meaningful indicator of the Satellite IMP status. Nevertheless, the stream count can still be obtained by typing a CONTROL-P in MONITOR. Another change made in RECORDER is that the BBN gateway is designated as the "preferred path" and not the "exception path" for Tanum.

Later in the quarter, we released Satellite IMP versions 3.5:1 for Honeywell 316s and 4.5:1 for BBN C/30s, which have Raisting, Germany, and not Clarksburg, Maryland, identified as



the fourth Satellite IMP site, although the Raisting site has not been installed yet. Necessarily, the host configuration tables and the host address tables needed changing. Correspondingly, we modified the TENEX/TOPS-20 SATNET monitoring programs RECORDER, MONITOR, QUERY, and EXPAK to process information from the Raisting site.

### 2.3 Hardware Maintenance Operations

During the last quarter, several hardware problems appeared which we diagnosed and, when they were related to the Satellite IMPs, fixed. In April, a major problem occurred marked by severe channel degradation, such that SATNET was for all practical purposes unusable for an entire week. The Satellite IMPs lost reservation synchronization repeatedly throughout the day, interrupting both gateway traffic and ARPANET Line #77 (the ARPANET direct connection via SATNET for London IMP connectivity). Consequently, the London IMP was declared down more often than up. Although initially Hello packet reception seemed only marginally affected, channel request packet reception showed many instances of high lossage.

First, we verified the software at all sites. Next, we determined that isolating Tanum (by crashing it) did not improve the situation. As a desperate expedient, we switched the network

from CPODA to FPODA, and the problem seemed to ameliorate; hence, we inferred contention packet resolution was failing. Since no changes have been made in this software for some time, we believed the problem was caused by a malfunction in the packet acquisition hardware. Not long thereafter, Hello packet reception deteriorated rapidly. Only afterwards did COMSAT determine that an unauthorized signal 12 db down and 500 Hz away from the SPADE pilot signal was present and that mismatched power levels were present at all sites. Both of these items were serious enough to create havoc with SATNET operations. Only after the interfering signal was identified and removed, which took several days, could the power levels be adjusted at all sites to normalize system operation.

In another major problem, a malfunction at Goonhilly created serious disruptions to SATNET service. To ascertain whether the software was corrupted, we reloaded the Satellite IMP (the very nature of the problem did not allow us to do a direct verify on the software in the machine); however, no improvement in operations was seen. Site personnel were then called upon to invoke various loopbacks in the system and to interrogate the Satellite IMP computer to determine whether packet lossage was occurring. Since the results of these tests showed normal packet reception when the Satellite IMP initiated an external crosspatch and impaired packet reception when the PSP terminal was set to

the data loopback state, we concluded that the malfunction originated in the PSP terminal hardware. COMSAT eventually corrected the problem, but it is not clear what steps worked; pressing the master reset switch on the PSP terminal failed to restore service. Only after the Goonhilly PSP terminal cards were reseated and resets on specific modules were pressed was service restored. Relevant to this problem is that on February 2 (reported in the last Quarterly Technical Report), the modem was moved from the A channel to the B channel of the PSP terminal in response to a B-channel modem malfunction and an A-channel connector malfunction.

We conjecture that the Goonhilly PSP terminal malfunction caused the unexplained problems seen on several occasions during the last quarter. Problem symptoms at the time suggested either a channel malfunction or a software bug. In order to ascertain whether the new software release was the culprit, we restored the previous software version 3.4:1 (this version had been in the network from last summer to last February). Because the symptoms were still present, we concluded that the problem was common to both the old version 3.4:1 and the new version 3.5:0. Since Tanum had on several occasions reported a different number of streams than Goonhilly and Etam, we also began looking carefully at the stream handling procedures. In recent times, the facility for providing automatic stream service for gateway traffic has

been used extensively, placing greater stress on the stream synchronization code as streams are created and deleted frequently in response to interactive traffic demands. Moreover, in this quarter, Telnet users at NORSAR have added to the demands on the facility, which we interpreted as a possible reason that the problem surfaced only recently.

Because analysis of the software revealed no problems, we had the BBNCC field service remove the Satellite IMPs from service and perform thorough hardware tests, including power supply voltages, memory, and I/O interfaces, on the Honeywell 316s at Tanum and Goonhilly. Both machines passed all tests and required no adjustments; hence, we concluded the SATNET difficulties were probably not attributable to a Honeywell 316 hardware malfunction. We hope that the corrective action taken on the Goonhilly PSP terminal has finally eliminated the problem.

Other hardware problems occurring in SATNET are as follows. The Goonhilly Satellite IMP abruptly began initiating power-fail/restart recoveries several times an hour. Site personnel determined that the cooling fans on the Honeywell 316 power supply had seized; when the power supply heated up, the power-fail/restart circuitry was triggered. To prevent heat damage, site personnel powered the machine off until three fans from the spares kept in London were sent to Goonhilly. Once the fans were

installed, normal service was restored.

A system failure in a New York TELCO office disrupted landline communications between Cambridge and Washington, D.C. During this period, the VDH circuit between the BBN gateway and the Etam Satellite IMP was down, isolating the entire European internet community and preventing SATNET monitoring reports from reaching RECORDER. ARPANET Line #77 was operational, though, providing London IMP connectivity. This failure also isolated the Clarksburg Satellite IMP. In a later unrelated incident, the VDH circuit between the BBN gateway and the Etam Satellite IMP failed again, thereby isolating SATNET, until TELCO fixed the problem.

A tractor severed the 50 Kb/s circuit between the UCL Gateway and the Goonhilly Satellite IMP, disrupting communications for several days. British TELECOM was called to fix the broken circuit.

A malfunctioning SPADE down converter at Etam caused deleterious reception problems at that site, interfering with SATNET channel service. Site personnel diagnosed and fixed the problem.

On several occasions modems in the 9.6 Kb/s circuit between the Goonhilly Satellite IMP and the London IMP entered into an

unknown state creating a one-way circuit. Following usual procedures, the corrective action taken was for Goonhilly site personnel to loop and immediately thereafter unloop the modem.

A lengthy outage in the London IMP connectivity to the ARPANET via SATNET resulted when the SDAC IMP was powered off to allow the installation of a new power generator at the site. Because the ARPANET site coordinator had been seriously ill, only a few minutes' advanced warning of the outage was given to the NCC and subsequently to the European users of the London IMP. Problems with the installation caused several subsequent outages with Line #77 at various times. Later, an unrelated hardware problem developed with the SDAC IMP, creating another service disruption and requiring BBNCC hardware maintenance personnel to repair the hardware.

### 3 PLURIBUS SATELLITE IMP DEVELOPMENT

The major activity during the quarter continued to be the support of Wideband Network system integration. The pace of this integration picked up with the establishment of June 3 as the date for a major demonstration of the system. In order to achieve this goal, every attempt was made during March and April to make the satellite subsystem available each Monday, Wednesday, and Friday for checkout and debugging of packet speech equipment at the Wideband experimental sites. BBN took on responsibility for generating and distributing reports summarizing the results of the experimental activity each day. The following paragraphs summarize the major operational problems encountered and milestones achieved during the past three months.

During February, progress toward satellite subsystem integration was frustrated by a number of equipment problems. There were PSAT hardware problems during most of the first half of the month at DCEC and during the second half of the month at ISI. At Lincoln Laboratory, the ESI was only semi-operational from 5 February until 18 February. In addition, there were thermal problems with the earth terminal at ISI early in the month. In spite of these site outages, there were several major accomplishments in February. On 3 February, a packet speech call via miniconcentrator gateways at ISI and Lincoln Laboratory was

supported for two hours. While such a call had been established before, this was the first instance where the system was robust enough for sustained operation. This same day the first multisite voice call between a Voice Funnel (at Lincoln Laboratory) and a miniconcentrator gateway (at ISI) was also demonstrated. Lincoln personnel installed the Packet-to-Circuit Interface (PCI) at DCEC on 10 February and after the DCEC satellite subsystem came up on 16 February, it was possible to establish a conversation between a packet voice terminal at Lincoln and a PCI supported telephone at DCEC. During the last week of the month, all three sites were operational on the satellite channel for the first time and ISI began initial experiments with packet video transmission.

Problems with the satellite subsystem continued to limit the progress toward stable network operation in March. Starting early in the month, both DCEC and ISI experienced intermittent periods of unusually high packet error rate coupled with error indications generated by the ESI. Interactions with Linkabit indicated that these problems were likely associated with a generic grounding problem that had recently been identified and was in the process of being corrected. Several brief outages of the ISI earth terminal also occurred during the month. These outages were associated with an improperly set trip level on the HPA helix current. The solution to this problem involved



operation with the 75 watt HPA until the end of the month while the 125 watt unit was taken out of service for adjustment. The DCEC PSAT was down during the period 26-29 March due to a burned out circuit breaker. A minor but longstanding bug blocking our ability to load the DCEC PSAT via the EDN gateway was traced in March to an incorrectly computed internet header length. This could be fixed simply by the distribution of a new set of PSAT host port loader paper tapes. Based on discussions at the Wideband meeting at DARPA, BBN sent Linkabit a list of known or suspected ESI bugs on 3 March. In addition, tests at higher symbol rates (i.e. above 772 Ksymbol per second) with coding enabled were carried out in order to provide Linkabit with additional information on performance of existing ESIs.

A three-site full-connectivity speech scenario was supported for the first time on 4 March. On 9 March the network was cutover to operation with the new Wideband Network addressing scheme. However, multisite experiments during much of the month were not possible due to the problems described above and integration efforts were, therefore, largely associated with local (single site) testing. On 24 March the SRI PSAT was installed and connected to the ARPANET via an IMP port expander. The port expander interface had been worked out earlier in the month at Lincoln Laboratory by BBN. Attempts to loop the PSAT through the SRI ESI on 25 March, however, indicated a problem

that required shipment of the ESI back to San Diego for repair. On 30 March, the second Voice Funnel was installed at ISI. Finally, on 31 March the first (local) voice conference was established between two packet voice terminals on two Lexnets connected to the Lincoln PSAT via two miniconcentrator gateways. During the month it was demonstrated that the low-speed host interface on the PSAT (HST) could be configured to run at approximately 600 Kbps (burst rate) rather than 300 Kbps as previously thought. This should greatly facilitate some of the experimentation currently planned by Lincoln personnel.

During April, Western Union began the installation of its remote monitoring and fault diagnosis equipment at the various sites. While this process was expected to be carried out with very little disruption to Wideband Network operation, in practice this was not the case. The new Western Union equipment was installed initially at DCEC during the week of 5 April and at Lincoln Laboratory during the week of 12 April. In both cases the installation caused the sites to be off the air for considerable portions of the week. Although the equipment appears to provide the proper status and alarm information to Western Union's Glenwood operations facility, it is not yet possible to obtain this earth terminal status in the PSAT via requests to the ESI. This problem should be addressed in the near future jointly by Linkabit and BBN. Linkabit began the

process of upgrading and standardizing their deployed ESIs in April. In general, these units were tested with the ISI PSAT prior to shipment to their destination site. The SRI ESI was returned to SRI and checked out with the SRI PSAT on 15 April. On 18 April the SRI site was brought up on the channel for the first time with ISI and on 21 April the first PCM conversation between packet voice terminals on Lexnets at the two sites was established. A failure of the Lincoln ESI on 13 April caused a shuffling of several units resulting in an upgraded ESI at Lincoln Laboratory and DCEC left without an ESI. Therefore, DCEC was unavailable for any network experimentation for the second half of April. On 9 April, the first multisite packet speech conversation supported via Voice Funnels at each end was carried out between Lincoln Laboratory and ISI. Finally, April was the first month when LPC as well as PCM packet speech began to flow over the Wideband Network.

In addition to supporting Wideband Network integration activities, work during the quarter has focused on various software enhancements to the PSAT and startup of the BSAT development. A new PSAT stream synchronization mechanism has been mentioned in the previous Quarterly Technical Report. Implementation continued on this project. A detailed description of the stream sync mechanism is included in section 3.1 of this report. Another development that was completed during the

quarter is the new host support software which takes advantage of the SuperSUE poller. This PSAT enhancement is described in section 3.2.

BSAT development activity during the quarter has primarily focused on the specification of a new BSAT/ESI interface and the initial design of the BSAT software architecture. A draft proposal for the new interface design is nearly complete and will be distributed to Linkabit and ARPA for review in the near future. The major changes in this proposed interface relative to the current interface design are:

Level 1: Multiple physical ports for expandability and reliability.

Level 2: HDLC burst framing.

Level 3: All precisely timed burst transmission and burst timestamping done by ESI.

BSAT does burst rearrangement to reduce the number of coding/modulation states per burst.

Elimination of CRC on individual data packets.

It is expected that there will be considerable discussion on both the philosophy and details of the BBN proposal over the next several months.

The conversion of the PSAT software from Pluribus assembly language (RATMAC) to Butterfly C code requires a certain amount of redesign to take advantage of the Butterfly hardware architecture and operating system (Chrysalis) environment. The Butterfly differs from the Pluribus in that processes rather than strips are the basic programming elements. Each process generally runs on a processor node specified by the system designer. The first step in the design of the BSAT has been to develop an initial allocation of controller functions into processor nodes. The functions generally correspond to existing modules of PSAT software although some BSAT-unique functions exist as well. The functional division among processor nodes attempts to equally partition the processing load and minimize the need for data transfers across the switch. The aggregation of functions into processes within a processor node attempts to balance the advantages of a more modular design with many concurrently executable processes against the advantages of a more efficient implementation based on a smaller number of processes. During the quarter, the BSAT software structure began to take form based on the above considerations and on numerous discussions with the Voice Funnel development group.

### 3.1 PSAT Stream Synchronization Algorithm

Stream synchronization refers to the maintenance of a distributed stream database to support network stream scheduling. It is essential for all PSATs to have identical scheduling information for the distributed satellite channel scheduling algorithm to operate correctly. After providing some background information, this section describes two approaches for stream synchronization which were considered. Advantages and disadvantages of each are identified, revealing a clear choice for the initial PSAT implementation. Finally, the fundamental operation of the selected stream synchronization algorithm is described in detail.

#### 3.1.1 Background Material

The distributed PODA operation of the Wideband Network requires all network PSATs to have consistent database information for satellite channel scheduling. For stream traffic, each PSAT must have accurate information regarding the channel allocations for all streams existing in the network. This information, which must be consistent network-wide, is referred to as channel stream data. Specifically, a channel stream refers to the dedicated satellite channel time for stream

traffic originating from a particular PSAT. In addition to channel streams, there are also host streams. A host stream refers to the dedicated allocation requested by a host to support its stream outgoing traffic. All host streams originating at a PSAT which are compatible with respect to certain parameters are aggregated into a single channel stream, resulting in a reduction in the total network overhead.

A host stream database is locally maintained by each PSAT. The information contained in this database need not to be known by all PSATs to schedule stream traffic. Channel stream data, on the other hand, must be known globally in the network. It is this channel stream information which must be maintained as a consistent database among all network PSATs for the distributed satellite channel scheduling algorithm. More information on host and channel streams can be found in the PSAT Technical Report, BBN Report No. 4469.

The distributed PODA protocol for creating, deleting and changing channel streams is designed to be reliable. Execution of database modification commands is globally synchronized so that each command takes effect at precisely the same time in every PSAT. Control messages affecting the stream database are sent multiple times in successive PODA frames to minimize the probability that a PSAT misses a database update (hereafter

referred to as a setup, stream operation, or stream op). Even so, it is necessary to detect and recover from database inconsistencies caused by a failure of all PSATs to successfully process setups. This is facilitated by periodically broadcasting the number of stream operations that have been processed. A PSAT can detect that it has missed a stream op if a broadcasted count is higher than its own. In the Wideband Network, the leader PSAT broadcasts its stream op count in every leader packet (i.e. every PODA frame). Other PSATs take the leader's count as the correct value and compare it to one maintained locally. If the local count differs from the leader's count, the PSAT is out of stream synchronization and needs to reacquire it.

There are two different situations when the accuracy and consistency of the channel stream database are uncertain. First, a PSAT initially coming up in the network has no stream database. As part of the initial acquisition procedure, the PSAT must obtain this database from another PSAT before becoming fully synchronized with the network. The second case is when a PSAT fails to receive a database update (stream create, delete, change) which is received and processed by the leader PSAT. Thus, the database information is inconsistent with that of the leader and is detected by a stream op count discrepancy. The PSAT missing the update must obtain the updated database information from the leader PSAT before it can accurately



schedule satellite channel time. Until the database is corrected, all data transmissions from the PSAT must be inhibited since the inability to correctly schedule streams precludes the correct scheduling of datagrams as well.

The algorithms which maintain a consistent stream database among all network PSATs, detect an inconsistent database, and recover from this condition are collectively referred to as stream synchronization.

### 3.1.2 Design Approaches

Two different design approaches were considered for the stream synchronization algorithms. The "active" approach requires a PSAT out of stream synchronization to request help from the leader PSAT. Responding to the help request, the leader sends the stream database over the satellite channel. The "passive" approach involves continual transmission of parts of the stream database in each channel stream burst. A PSAT which is out of stream synchronization simply listens to the received stream bursts to (re)construct its channel stream database. Each of these approaches is now discussed in more detail.

### 3.1.2.1 The Active Approach

When a PSAT first detects that it is out of stream synchronization, it immediately transmits a special stream help packet. A help packet received by the leader PSAT results in the transmission of a stream database datagram. This datagram contains sufficient information on all channel streams to either build the database from scratch, in the case of initial acquisition, or verify and correct the database during recovery. The acquiring PSAT uses this datagram message to (re)construct the stream database as of some time in the past. To bring the database completely up to date, the PSAT must then process all setups received since detection of the out-of-synchronization condition. Once all of the deferred stream setups have been processed, the PSAT is in stream synchronization.

### 3.1.2.2 The Passive Approach

When a PSAT first detects that it is out of stream synchronization, it begins to listen to all channel stream bursts received. Each channel stream scheduled is required to transmit a burst, regardless of whether data exists to send in the stream or not. The control portion of the burst is expanded to include sufficient information to reconstruct a channel stream database entry. This information is identical to that which would be sent

for the stream in the stream database message using the active approach. Since every channel stream is scheduled at least once every 8 PODA frames, stream synchronization may be (re)acquired in as little as 8 frames.

As in the case of the active approach, stream setups received while in the out-of-stream-synchronization state must be processed in a deferred fashion. However, the passive approach necessitates much more complex processing. In the active approach, the acquiring PSAT is supplied a snapshot of the stream database frozen at the time of database message creation. In the passive approach, no frozen snapshot is provided. The database contents may change while processing successive channel stream bursts. As a result, more extensive bookkeeping must be performed. In addition, setup bursts must be expanded to include the same information included in the stream bursts. This permits a PSAT to track changes occurring to the database as it tries to (re)construct it. Once this (re)construction is complete and all deferred stream setups have been processed, the PSAT is in stream synchronization.

### 3.1.2.3 Comparison of Approaches

Both the passive and active approaches have advantages and disadvantages. However, the relative advantages of the active

approach are compelling. First, the passive approach requires the continual transmission of stream synchronization information. Channel stream information must be included in all channel stream bursts and setups. This additional network overhead occupies satellite channel bandwidth which could have been used for user data transmission. The active approach entails no such overhead. Second, the algorithms of the passive approach are much more complex than those of the active approach. Specifically, the (re)construction of a dynamically changing database requires extensive bookkeeping and cross-checking. The active approach, with its snapshot of a static database, is quite simple by comparison.

The advantage of the passive approach is speed. Stream synchronization can be (re)acquired in as little as 8 PODA frames or approximately 160 milliseconds. The active approach requires at least 3 round trip times (help packet, datagram reservation, and datagram) or 750 milliseconds. A shorter (re)acquisition time translates into a shorter speech drop-out period. However, this advantage disappears when the case of dangling streams are considered. In particular, an out-of-synchronization PSAT which owned streams prior to losing synchronization must recover the necessary information about its streams before reacquiring synchronization. However, no PSAT is transmitting such information over the satellite channel. Either the acquiring

PSAT requests this information from the leader or it is forced to delete all its streams. The latter is obviously inadequate. The former involves a delay of three round trip times and reduces to essentially the active approach.

With the above factors in mind, the initial implementation of stream synchronization is that the PSAT will use the active approach. The following section discusses the fundamental operation of the PSAT synchronization algorithms.

### 3.1.3 Fundamental Operation

The (re)acquisition of stream synchronization in the PSAT consists of five distinct phases. These phases are described in the following sections.

#### 3.1.3.1 Stream Database Acquisition Phase

Assuming the PSAT is in frame synchronization (tracking network global time and PODA framing boundaries), the PSAT first enters the stream database acquisition phase. During this phase, the PSAT requests the stream database from the leader PSAT. Processing of further stream ops is deferred until an accurate database on which to execute the stream ops has been

(re)constructed. Requests for the stream database are sent in the PODA control subframe using a special Stream Help Request packet. The sending PSAT transmits this request until it correctly receives the packet, to increase the chances that an error-free copy has been received by the leader PSAT. If this packet is not successfully received by the sending PSAT in approximately one round trip time, the PSAT will retransmit the packet. After 8 unsuccessful attempts to receive its own Stream Help Request, the PSAT will reinitialize. Upon successful reception of the request packet, the PSAT waits for a 2.5 second time-out period. If a stream database message is not received within this period, the PSAT will reenter the stream database acquisition phase and retransmit a Stream Help Request packet.

When the leader PSAT receives a Stream Help Request, it creates a datagram message containing its channel stream database and transmits it. The datagram is not addressed to a particular PSAT; any PSAT in the process of stream synchronization acquisition can use the information contained in it. The channel stream database datagram contains a creation timestamp (PODA frame number), current stream op count, total number of channel streams in the database, and channel stream data for each channel stream that exists. For each channel stream, five words of information are sent in the message: the channel stream ID and four words of allocated channel time for each of the four network

priority levels. The order of channel stream entries in the message is chronological by creation time.

### 3.1.3.2 Stream Database Build Phase

When the PSAT attempting to acquire stream synchronization receives a stream database message successfully, the PSAT enters one of two phases depending on whether the PSAT is initially coming up or recovering from a stream op count discrepancy. If it is first coming up on the network, it must construct the stream database from scratch and enters the stream database build phase. The stream database message contains enough information for the creation of a database identical to the one maintained by the leader PSAT at the time the stream database message was created. The PSAT performs a sequence of stream create operations, creating database entries for every channel stream described in the received message. This is possible since the order of channel streams in the message is arranged according to the original order of stream creation in the network as noted above.

### 3.1.3.3 Stream Database Recovery Phase

If the PSAT is recovering from loss of stream synchronization due to a stream op count mismatch, the stream database recovery phase is entered. In this phase, the PSAT performs a comparison of its stream database to the one contained in the database message. Any discrepancies found are immediately corrected if possible, or marked for further action. If the discrepancy discovered is for a channel stream belonging to another PSAT, the database can be immediately corrected by locally changing, creating, or deleting an appropriate channel stream entry. If the discrepancy discovered is for a channel stream belonging to the acquiring PSAT, correction is more complex. Three separate cases are possible.

For the case where the leader PSAT database contains an entry not present in the database of the acquiring PSAT, the channel stream must be globally deleted. The local host stream structure making up the channel stream cannot be reconstructed, so that the channel stream is unusable in its current state. The channel stream cannot be immediately deleted, however, as the database must be completely recovered before any modifications are made. As a result, the stream is marked for deletion in the subsequent database clean-up phase.

For the case where the leader PSAT database does not contain



an entry present in the database of the acquiring PSAT, the channel stream entry is locally deleted and the corresponding hosts notified of the deletion. Sufficient information is present to try to globally create the channel stream in order to add it to the leader's database. However, this stream creation could only be attempted after the database has been fully recovered or during the database clean-up phase. Even then, resources may not be available to create the channel stream. Because of the additional complexity of carrying out this stream creation process, the initial implementation of PSAT stream synchronization has taken the simpler approach based on deletion. Should this prove inadequate, the more sophisticated approach can be implemented.

Where corresponding channel stream entries in both the acquiring PSAT and leader databases exist, the discrepancy must be in the slot sizes describing the stream allocation. Here again, sufficient information exists in the PSAT to attempt to globally change the channel stream to match the local slot sizes. This operation would have to be attempted in the database clean-up phase when the database has been completely recovered. However, a much simpler alternative is to delete the channel stream completely. Since the existence of only a small number of host streams per channel stream is anticipated in the near future, the deletion of the channel stream should rarely cause a

problem to the user. If this deletion approach proves inadequate, the additional functionality to globally attempt a change in the stream's allocation can be implemented at a later date.

#### 3.1.3.4 Deferred Setup Processing Phase

Once the stream database build or stream database recovery phase is complete, the PSAT enters a phase in which it processes deferred setups. Going into this phase the PSAT has constructed an accurate database corresponding to some time in the past. All setups received while in previous stream synchronization phases have not been processed but have been placed on a deferred setup queue. In the deferred setup processing phase of stream synchronization, those setups are now processed to bring the stream database up to date. Actually, it is only necessary to process those setups which are marked for execution in or after the PODA frame in which the stream database was created since other setups are already reflected in the database itself. Setups are dequeued from the pending queue and executed in FIFO order updating the stream op count appropriately. Once this setup processing is complete, the PSAT is in stream synchronization and can begin scheduling stream traffic again. We note that if the PSAT always keeps one round-trip time of

previously received stream setups available, it is possible to process database messages sent by the leader in response to Stream Help Requests from other out-of-sync PSATs as well as direct responses to local help requests.

### 3.1.3.5 Stream Database Clean-up Phase

The final phase of stream synchronization is basically a clean-up phase. In the case of initial acquisition, there may be outstanding streams for this PSAT, which the other PSATs have a record of, but which were destroyed locally by reinitialization. When the PSAT comes back on the network, all host stream information has been lost. The channel stream information still exists, however, and is provided to the PSAT in the stream database message. As noted above, it is necessary to have the PSAT clean up these outstanding streams by globally deleting them. To do this, the PSAT keeps track of all channel streams owned by it when processing each entry of the stream database message. After the deferred setup processing phase has been completed, the PSAT globally deletes these streams one by one. As a result, all streams existing prior to the reinitialization must be reestablished by normal host requests.

In the case of recovery from loss of stream synchronization, the clean-up phase is identical. Discrepancies between the

leader's database and the local PSAT database for channel streams owned by this PSAT may have been only partially resolved during the stream database recovery phase. Remaining discrepancies are resolved by deleting the channel stream in an identical manner to that performed for initial acquisition.

### 3.2 PSAT Host Support Software

The PSAT has three device types: HST, HSM and SMI. The HSM (High-Speed Modem) is used for hosts which need a high speed link to the PSAT, such as the Voice Funnel. HSMs generally need to be serviced more frequently than HSTs to prevent latency problems. The HST is used as an interface to hosts which generally support lower levels of traffic and do not have critical time constraints. Since HSTs can block hosts if they are not serviced, they are not susceptible to latency problems. The SMI is the interface between the ESI and the PSAT. It is critical that the SMI must be serviced rapidly enough to keep up with the 3.088 megabit per second satellite channel. Section 3.2.1 describes the host support software structure as it has existed for some time. Section 3.2.2 describes the new host software that will be installed in the near future.

### 3.2.1 Old PSAT Host Software

Previously, there were several types of pollers used to drive the various devices in the PSAT. Each of these pollers corresponded to a different tradeoff between device latency and applied processing power. The PSAT application used the HSTPOL routine to drive both the transmit and receive side of the HST devices. A variation of HSTPOL, HSMPOL, was used to drive the HSM devices. Both HSTPOL and HSMPOL operated within the PSAT application. They were designed to service the HST and HSM devices whenever there was input or output. Since the SMI is required to keep up with the satellite channel, it must be driven by a dedicated poller operating outside of the general application program. The P13 Poller was originally configured for this purpose. The P13 Poller is actually one of the Pluribus processors, which is removed from running the application and dedicated to running a loop to poll the SMI. Because the P13 Poller took away needed processing power from the PSAT and was not always capable of keeping up with the satellite channel, the SuperSUE Poller (SSP) was built. The SSP is a separate device dedicated to polling other devices for messages. Each SSP fits into one of the slots on an I/O bus, and can be used to poll one or two devices located on that bus.

The PSAT is configured with two I/O buses. the E-Bus and the

F-Bus. Each I/O bus has one HSM. Currently, the Voice Funnel uses only one of these HSMs and the other is turned off. Each PSAT also uses only one SMI, which must be on the same I/O bus as an SSP in order for the SMI to be polled by it. The SMI on the other bus is kept as a spare.

Until recently, the PSAT was told whether or not to use the SSP by the value set in the SSUE flag, and where the SSP was located by the value set in the SSADR variable. After the SMI was discovered, the PSAT looked at the SSUE flag to determine whether or not to use the SuperSUE Poller. If the SSUE flag was set, the SSP was initialized to poll the SMI. If the SSUE flag was not set, the P13 Poller was activated. The code assumed that the discovered SMI and specified SSP were on the same I/O bus. When the HST and HSM devices were discovered, they were always initialized to be polled by the HSTPOL and HSMPOL routines respectively. This was wasteful of both the ability of the SSP to poll two devices and the processing power of the Pluribus which was diverted from the application to run HSMPOL. It also disallowed the use of multiple SSPs.

### 3.2.2 New PSAT Host Software

The new host code was built to allow the PSAT to discover all its I/O devices properly and to initialize the discovered

devices to use the pollers in the most efficient way. In the process, the host I/O routines responsible for polling (HSTPOL/HSMPOL) and the message level processing routines (MSGIN/MSGOUT) have been modified to be more efficient.

The major modification to the polling routines is that the HSTPOL and HSMPOL routines have been replaced by HINPOL and HOUTPOL. These two new routines also run within the PSAT application. HINPOL and HOUTPOL can be used to poll both HSTs and HSMS. HINPOL is invoked by the device whenever it has input to be processed; HOUTPOL is invoked by the PSAT application whenever it has output to be transmitted by the device. Because of constraints associated with maintaining the host link, HOUTPOL must also be invoked once per second to reset the ready status of the device (kept in the device transmit status register). It is hoped that splitting up the input and output sides of the poller will allow the device to be serviced more efficiently.

The MSGIN and MSGOUT routines basically maintain the same functionality as before. They are responsible for breaking up messages into buffers on output, and reassembling buffers into messages on input. When the SuperSUE Poller polls a device, there are constraints as to where its buffer tables can be. Specifically, the tables for its device 1 and device 2 must be located in consecutive locations on the same memory page.

However, there can potentially be a large number of hosts. so the tables for hosts should, in general, be dynamically allocated. Therefore. MSGIN and MSGOUT were modified to choose either the dynamic structure used by HINPOL and HOUTPOL, or the static structure used by the SSP, depending on which poller is servicing the I/O device.

The new host code allows the application to discover at most one SSP, one SMI, and one HSM on each of the two I/O buses. Any additional devices of these types which are present in the PSAT are ignored. Each PSAT in the field will typically have two HSMs, two SMIs, and two SSPs present. The new host code does not constrain the number of HSTs which it can support. The user can choose which of its HST devices to use at any given time by turning them on or off in the table of permanent hosts (PERHST table).

The initialization and configuration of the discovered devices is done over a period of time due to constraints supplied by the PSAT operating system. The PSAT allows the application first to run routines to initialize the application ("initialization time"). There are also application routines for initializing individual devices as they are discovered ("device discovery time"). Once all the devices have been discovered, the application can use the information it has gathered to do any



further initialization and configuration which depends on knowing where all of the discovered devices are located.

Each HST which is turned on is found and initialized during device discovery time. HSTs are always serviced by the HINPOL and HOUTPOL routines. The SSPs, SMIs, and HSMs pose a different problem when it comes to device initialization and configuration. With both the SMIs and the HSMs, the PSAT needs to know where the SuperSUE Pollers are. However, this information cannot be known until the end of device discovery time. Consequently, the routines run during initialization and device discovery time merely store away the locations of devices of each type. All of the real initialization and configuration is done later.

As noted above, each PSAT can have at most one usable SSP on each of its I/O buses. Whether or not these SSPs are discovered depends on the setting of the SSUE flag. If the SSUE flag has been cleared, the SSPs present will not be discovered. Each SSP that is discovered can poll a maximum of two devices, both of which must be on its I/O bus. Each will poll at most one HSM and one SMI. If it polls an SMI, the SMI will always be device one to the SSP. Similarly, an HSM will always be device two.

Although the PSAT normally has two SMIs, one on each I/O bus, it only uses one of them, keeping the other as a spare. After discovering the SMIs, the PSAT must intelligently choose

which to use, based on the locations of the known SuperSUE Pollers. It is preferable to have the SMI polled by an SSP if possible. The PSAT will first try to find an SMI and an SSP on the same bus. If there are any such pairs, it chooses one. If it finds none, it chooses one of the SMIs it found, and enables the P13 Poller.

Table 1 shows the possible locations of discovered SMIs and SSPs with the resulting choices made by the PSAT as to which SMI is running and which poller is attached to it. SMIs can be present on either the E or F I/O buses, or on both. Similarly, there can be SSPs on both the E and F buses, on one of the buses, or on neither bus. Each table entry is of the form SMIx/Poller (x = E or F). SMIE indicates the SMI on the E-Bus, and SMIF indicates the SMI on the F-Bus. The two possible SuperSUE Pollers, SSPE and SSPF, are on the E and F buses, respectively. P13 is the P13 Poller.

A PSAT will have at most one HSM on each of the two I/O buses. The user can control which HSMs are discovered by turning them on or off in the PERHST table. At the end of initialization and device discovery time, the PSAT knows where the usable HSMs and SSPs are. The PSAT will then decide which HSMs will be serviced by which SSPs, and which will be serviced by HINPOL and HOUTPOL. If the discovered HSM is on the same I/O bus as a

discovered SSP, the SSP is initialized and the HSM is set up to be polled by that SSP. Otherwise, the HSM is configured to be serviced by HINPOL and HOUTPOL.

Table 2 shows the possible locations of discovered HSMs and SSPs. There can be HSMs on either the E or F I/O buses. or on both. Similarly, there can be SSPs on both the E and F buses. on one of the buses. or on neither bus. In each case, the table shows which poller is running which HSM. Each table entry is of the form HSMx/Poller (x = E or F). HSME indicates an HSM on the E-Bus, while HSMF indicates an HSM on the F-Bus. SSPE and SSPF are again the two possible SuperSUE Pollers. HPOL is an abbreviation for the HINPOL and HOUTPOL routines.

SMI LOCATIONS				
S S P L O C A T I O N S		E-Bus, F-Bus	E-Bus Only	F-Bus only
	E-Bus F-Bus	SMIE/SSPE	SMIE/SSPE	SMIF/SSPF
	E-Bus only	SMIE/SSPE	SMIE/SSPE	SMIF/P13
	F-Bus only	SMIF/SSPF	SMIE/P13	SMIF/SSPF
	None	SMIE/P13	SMIE/P13	SMIF/P13

Table 1.

HSM LOCATIONS				
S S P L O C A T I O N S		E-Bus, F-Bus	E-Bus Only	F-Bus only
	E-Bus	HSME/SSPE	HSME/SSPE	HSMF/SSPF
	F-Bus	HSMF/SSPF		
	E-Bus only	HSME/SSPE HSMF/HPOL	HSME/SSPE	HSMF/HPOL
	F-Bus only	HSME/HPOL HSMF/SSPF	HSME/HPOL	HSMF/SSPF
	None	HSME/HPOL HSMF/HPOL	HSME/HPOL	HSMF/HPOL

Table 2.

#### 4 REMOTE SITE MAINTENANCE

This quarter was one of transition for the Remote Site Maintenance (RSM) project due to personnel changes at both BBN and the ACCAT sites. Routine RSM work continued to provide a good level of support for the operational sites.

Development work centered on an accounting system for the NOSC ACCAT site, configuration and testing of a device driver to provide terminal modem control, and conversion of the GL network graphics backend protocol module, developed by RAND, for use with the Genisco graphics microcode in use at the RSM sites.

##### 4.1 RSM Accounting

The UNIX system does not naturally support accounting. The standard distribution from Western Electric contains some features to collect and analyze accounting data, but does not have a clear-cut concept of an account. The only available units which can be used are the "user" and the "group". We have chosen to use the "group" as the basic accounting unit, though this does cause some interference with the file protection system, which assumes that groups are related in an entirely different way. Persons who are working closely together may be charging their activities to different jobs, and will find that they trip over

the protection mechanism from time to time. The only way to avoid this would be to provide a new identifier which specifies the job number.

The UNIX operating system as distributed has a mechanism for collecting accounting data for each process as it terminates. The data includes user and system process time, elapsed time, user and group IDs, and the command name. The raw accounting data is rather bulky. Each transaction occupies 32 bytes, and there are a number of processes generating transactions even when the system is quiescent. On the other hand, it is necessary to keep the raw data around for a while to resolve questions which may arise about possible billing errors. To prevent the accounting data file from becoming too large, and to provide some isolation from disk errors, the raw data is backed up nightly to archival disk storage. In addition, each night a program is run to compute disk usage on a per user, per group basis. This program, blkent, is based on the UNIX du command.

For each period in the basic accounting cycle (at BBN this is semi-monthly), the raw accounting data is compressed, and reports are generated to create the actual billings. The sa program is used to convert the binary accounting data into readable form and do some compression. The standard UNIX version of sa has been modified in the following ways:

- o The original money calculation was based on user identification. This has been changed to support the group as the basic accounting entity. The information is accumulated for each group and user, and written out at the end of processing in a format which is suited to later processing by the report generator programs.
- o The top level shell is dropped from the accumulation, since minimal resources are consumed by an inactive shell and later correction of the records for the terminals left logged-in over night is complicated and expensive. In addition, certain other commands are dropped, among them login, date, getty, and stty. These commands often reflect spurious user/group identification.
- o Shift differentials have been added for use in the NOSC ACCAT system.

Once the raw accounting data has been digested, reports are generated by two awk programs which operate on the process accounting and disk usage data. The two reports include a summary of disk storage, process time (elapsed), and CPU time for each group, converted into computing resource units (CRUs), which are multiplied by a provisional rate to give an estimate of the billing. A more detailed report, broken down by user in each group, is also provided for use by individual project managers.

In summary. the accounting system keeps track of the following:

- o CPU time and elapsed process time. cumulative over groups and users.
- o Number of blocks of disk storage, cumulative over groups and users.
- o Billings, which are generated by applying a formula that



converts CPU time, process time, and averaged disk storage into CRUs, and includes shift differentials. .

- o Management reports, which are provided for each accounting group, and break the data down further to individual users.

This accounting scheme is based on one which has been in use in the BBN UNIX Cost Center for some time. It is being adapted for the NOSC ACCAT site by adding shift differentials and modifying the report generator programs. We expect to install this system at NOSC in the coming quarter.

## 5 INTERNET DEVELOPMENT

The major activity during the past quarter was the continued deployment and maintenance of the Macro-11 gateway. Other important work included operation and maintenance of the BCPL gateways, adding support for Packet Radio networks, enhancing the NU gateway monitoring, adding support for a fourth port at the UCL gateway, procuring a 9.6Kb line from Telenet, and definition of the "Stub" gateway protocol.

## 5.1 Macro-11 Gateway Installations

The Macro-11 gateway has been operational since early January. Three new sites (DCEC, BBN-PR, and SRI-C3P0) were added in the last quarter. The list of operational gateways is shown in the following table.

<u>Gateway</u>	<u>Adjoining Networks</u>
BBN-PR	BBN-PR/BBN-NET
BBN-NET	ARPANET/BBN-NET
NDRE	SATNET/NDRE-TIU - NDRE-RING
UCL	SATNET/UCLNET - RSRE-NULL
BBN	SATNET/ARPANET
SRI-C3P0	ARPANET/SF-PR-2
DCEC	ARPANET/EDN

## 5.2 New Gateway Features

The Internet Control Message Protocol (ICMP) was turned on in the Macro-11 gateways on March 1, 1982. The ICMP protocol is now used to send Destination Unreachable and Redirect instead of using the Gateway-Gateway protocol (GGP). Also, ICMP is used to send Echo Replies, Time Expired, Parameter Problem, and Information Request messages. Although the Macro-11 gateway will still respond to GGP Echo Request packets by sending Echo Replies, this feature is expected to be dropped in the future.

In the last quarter we added support in the Macro-11 gateways for Internet Class A, B, and C addresses. The gateways both understand the new formats and use them to exchange routing updates between the gateways. Following is an example of an output of a NU status command showing the network table of a gateway which includes the new address formats.

Arpanet	10	Directly connected
BBN-net	3	Directly connected
Fibernet	24	1 hop via DOS 10.1.0.49 (Arpanet 1/49)
BBN-TC	192.1.2	2 hops via FIBERNET 3.2.0.50 (BBN-net 2/50)
BBN-Jericho	192.1.3	2 hops via FIBERNET 3.2.0.50 (BBN-net 2/50)
Purdue-CS	192.5.1	2 hops via PURDUE 10.2.0.37 (Arpanet 2/37)
Wideband	28	2 hops via VAN 3.0.0.11 (BBN-net 0/11)
Satnet	4	1 hop via BBN 10.3.0.40 (Arpanet 3/40)
DCN-Comsat	29	1 hop via MILLS 10.3.0.17 (Arpanet 3/17)
Bragg-PR	9	1 hop via BRAGG 10.0.0.38 (Arpanet 0/38)
Lcsnet	18	1 hop via MIT-LCS 10.0.0.77 (Arpanet 0/77)
Uclnet	11	2 hops via BBN 10.3.0.40 (Arpanet 3/40)
Rsre-Null	35	2 hops via BBN 10.3.0.40 (Arpanet 3/40)
BBN-GT-TestC	192.0.1	1 hop via TIU 10.3.0.76 (Arpanet 3/76)
Ndre-Tiu	48	2 hops via BBN 10.3.0.40 (Arpanet 3/40)

Ndrenet	50	2 hops via BBN 10.3.0.40 (Arpanet 3/40)
Rsre-Ppsn	25	3 hops via BBN 10.3.0.40 (Arpanet 3/40)
Decnet-Test	38	3 hops via MILLS 10.3.0.17 (Arpanet 3/17)
EDN	21	Unreachable
DOSnet	192.1.6	1 hop via DOS 10.1.0.49 (Arpanet 1/49)

The mechanism to send Trap reports to the NU Network Operations System was changed in several ways. We had noticed that under certain conditions many duplicate traps were being sent by the gateways and that a gateway would attempt to send traps even though it did not have a route to the NU system. The following changes were made to the the Trap-sending module.

1. Traps are now sent every 10 seconds instead of every 1 second. That limits the maximum number of traps that a gateway can send to 8 (the current number buffered) per 10 seconds.
2. The Trap buffer is not cleared until the trap message has been accepted by the forward routine. This means the gateway will not send traps until it has a route to the NU system.
3. When the same Trap occurs more than once in a row. instead of being put directly in the buffer, a count is incremented. This count is sent along with the trap. Previously, the duplicate trap was just discarded.

The counts of the duplicate traps are now output by the NU system along with the Trap reports. This is very helpful to our

understanding of the number of times certain events happen in the operational gateways. An example of some trap output follows. The counts of the duplicate traps are at the end of a line in parentheses.

```

07:18 UCL 3 T1002: ENQ fail, flushed 35.7.0.0 -> UCL 35.7.0.0 (2.)
      UCL 3 T1031: HDLC - link up
      UCL 3 T2008: Interface UCL 35.7.0.0 up
      UCL 3 T2004: Neighbor RSRE 35.6.0.0 up
07:23 UCL 3 T1032: HDLC - link down
      UCL 3 T1033: HDLC - link conn. fail - 02
      UCL 3 T2001: Neighbor RSRE 35.6.0.0 down
07:24 UCL 3 T1002: ENQ fail, flushed 35.7.0.0 -> RSRE 35.6.0.0
      UCL 3 T1002: ENQ fail, flushed 35.7.0.0 -> UCL 35.7.0.0 (2.)
      UCL 3 T1031: HDLC - link up
      UCL 3 T2004: Neighbor RSRE 35.6.0.0 up
07:37 BBN 2 T2001: Neighbor UCL 4.0.0.60 down
      BBN 2 T1010: RDR: 10.0.0.27->11.2.0.42 via RCC 10.3.0.72 (42.)
      RCC 1 T1010: RDR: 10.0.0.27->11.2.0.42 via BRAGG 10.0.0.38 (45.)
      C3PO 14 T1010: RDR: 10.0.0.27->11.2.0.42 via RCC 10.3.0.72 (2.)
07:38 BBN 2 T2004: Neighbor UCL 4.0.0.60 up

```

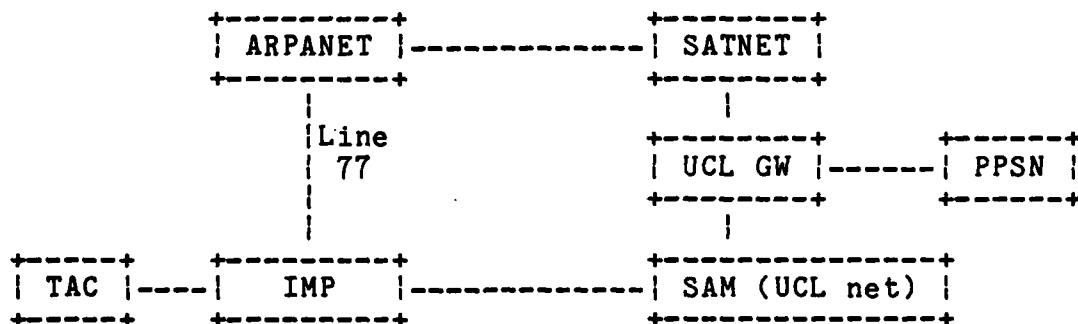
### 5.3 Packet Radio Gateway

A driver has been implemented to support the Packet Radio access protocols. The driver will support the CAP 5.6, CAP 6.1, and CAP 6.2 packet radio protocols. It was first tested with the BBN Packet Radio Network using CAP 6.2 protocol. Since then it has been installed at SRI in the C3PO Gateway. This has been running since it was installed. We are awaiting further testing by SRI; we are planning to test the other CAP protocols at SRI after the CAP 6.2 tests are complete. At that time, the Macro-11

gateway will be installed in the other Packet Radio gateways, which are the remaining BCPL gateway sites.

#### 5.4 UCL Gateway and UCL TAC

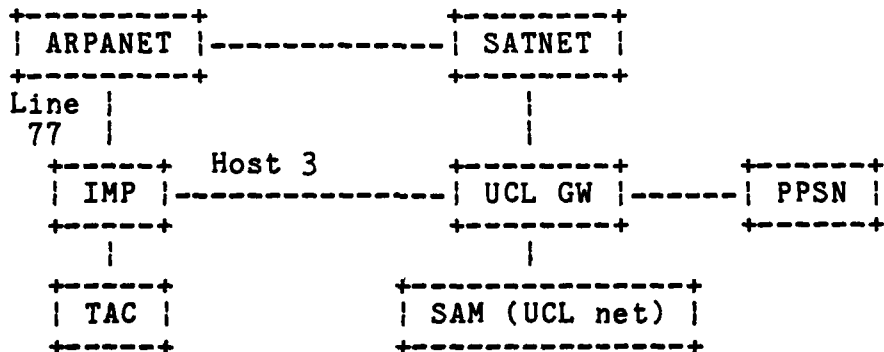
The London TIP at University College London (UCL) was upgraded to an IMP-TAC combination. This was accomplished by moving the Norway TIP, which was no longer on the network, to UCL. The TCP/IP portion of the TAC is now running as part of UCL-NET. All of the Internet traffic to the ARPANET is routed via the UCL gateway. The NCP portion is running as a host on the ARPANET. The current configuration is as follows.



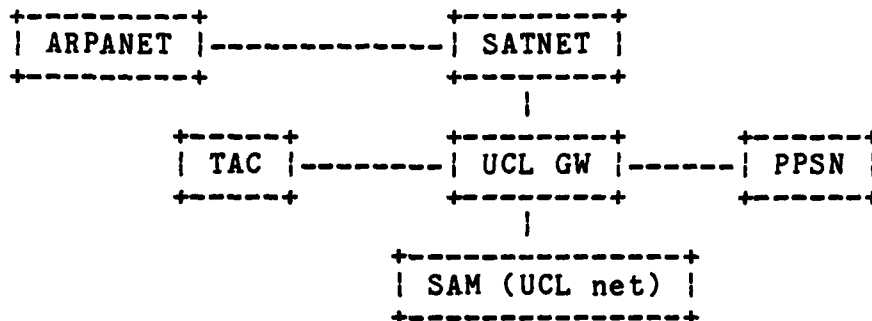
We plan to be able to remove the IMP and Line 77 and to install the TAC directly on the gateway. This procedure is described in two steps below.

1. The first step is to connect the IMP directly to the UCL

gateway. This involves adding a fourth 1822 interface to the UCL gateway and connecting it to the IMP's third host port. We have written a driver for an LHDH interface which is to be installed in the gateway. We are currently waiting for the LHDH hardware to be installed, in order to test the gateway with the new software. The TCP/IP portion of the TAC will be configured to run as a separate network, for which purpose it has been assigned network number 32. The NCP portion will continue to run on the ARPANET via Line 77. The gateway will be set up to use 256 byte buffers for all of its interfaces. The configuration for this step is as follows:



2. The final step is to remove the IMP and Line 77 and install the TAC directly on the gateway. This configuration is as follows:



All traffic from the networks at UCL to the ARPANET will go through the UCL gateway and, of course, must be Internet traffic.

#### 5.5 UWISC Gateway

A new gateway has been configured for the University of Wisconsin. It will be a two-port gateway: ARPANET 1822 to Proteon (V2LNI) Local Area network. The drivers (LHDH 1822 and V2LNI) for these networks had both been previously written. No major work was required. We are waiting for word to test out the gateway at Wisconsin.

#### 5.6 NU Monitoring Work

We have been working to improve the NU gateway support in two areas, Status monitoring and Statistics collection. Our



eventual goal in this work is to provide the tools to allow the Network Operations Center (NOC) to maintain the gateways in the same way as they now maintain the ARPANET.

Status monitoring involves building automatic processes to periodically find out the status of all of the gateways. This will be used provide information on which gateways are up or down, which of their interfaces are up or down, and if gateways and/or networks are up or down. This information will provide us with a global picture of the status of the Internet. Having an automatic means of informing an operator that a gateway is down is the first important step to permit 24 hour NOC support.

The first types of statistics we will be collecting involve measuring the performance of the gateway. This data will be collected over a time interval to enable us to understand how much traffic the gateways are supporting. The following types of data will be collected:

- o Interface Counters (collected for each interface)
  - Datagrams received
  - Data Bytes received
  - Datagrams sent
  - Data Bytes sent
  - Datagrams dropped
- o Global Counters
  - Datagrams addressed to gateway
  - Datagrams originated at the gateway
  - Datagrams dropped due to no route

o Packet Types Counters

- ICMP datagrams of each type received
- ICMP datagrams of each type sent
- GGP datagrams of each type received
- GGP datagrams of each type sent

We are also working on collecting data to build a Host Traffic Matrix (HTM), to find out how many datagrams are sent between the hosts in the internet. The gateways will count the number of datagrams sent between source and destination internet addresses. The data will be assembled in a table and will be sent when the table is full or a time interval has elapsed. This data should tell us which hosts send the most traffic through the internet and what the most utilized paths are.

#### 5.7 VAN Gateway and Telenet Line

After much discussion with the Telenet salesman a 9.6Kb host line for the VAN gateway work is on order and is scheduled to be installed at BBN about August 1, 1982. The line delivery is dependent on the ACC XQ/CP X.25 interface being certified by Telenet.

There are several important issues that still need to be resolved in operation of the VAN gateway between the ARPANET and UCL-NET. These are:

- o Which side (ARPANET VAN gateway or UCL VAN gateway) will be responsible for opening a Virtual Circuit (VC) to the other side?
- o If no VC is open, what is the mechanism whereby one can open a new one and still minimize packet charges for the ARPANET VAN gateway?
- o If a VC is not open, what should the VAN gateways do with datagrams destined to the other VAN gateway? Should they be discarded or held for some time?
- o How should the VAN gateway system be used to provide "back door" access for SATNET and Gateway debugging and who should pay for the packet charges for this work?

#### 5.8 Stub Gateway Protocol

We are currently working on designing a "Stub" Gateway protocol to allow different independent systems of gateways to interact and exchange routing. Our preliminary ideas on how the protocols should work has been described in messages presented to the Internet ICCB. We are now working on a document to describe the actual protocol.

#### 5.9 Internet Meeting

The Spring Internet meeting was hosted at BBN. Status reports on the current internet projects and talks on the Macro-11 gateway and on the NU gateway monitoring work were presented.

## 6 MOBILE ACCESS TERMINAL NETWORK

Our participation in the development of the Mobile Access Terminal (MAT) and the MAT Satellite Network (MATNET) during the last quarter was reduced to a low-level support while waiting for contract renewal. We did, however, participate via telephone conversations in the system integration satellite tests within the Advanced Command and Control Architectural Testbed (ACCAT) experiment at the Naval Ocean Systems Center (NOSC) in San Diego, California. Also, whenever postponement did not make sense, we performed some of the conversion of the MATNET Satellite IMP source software in parallel with the SATNET Satellite IMP source software to a form compatible with a C/30 cross assembler currently residing in BBN-UNIX machines.

## 7 TCP FOR THE HP3000

The last quarter of the HP3000 Internet project was spent on low level maintenance tasks of the HP3000 software. Very little time was actually charged to the contract because our software tests did not reveal any new bugs. This low level effort will continue until the C/30 IMP needed to connect the DARPA HP3000 to the ARPANET is delivered to DARPA. Delivery of the IMP will allow us to install our software at DARPA.

In addition to our software work we answered several queries about our TCP/IP software. These queries came from such diverse sources as NSF, the Air Force, Booze Allen Hamilton, Hewlett Packard, and the University of Delaware. To date we have not had any requests for our software.

We had one episode with the LSI-11 front end processor which we maintain under this contract. After some remote diagnosis of the system over the ARPANET we isolated the problem to the modem multiplexer which connects the LSI-11 to an HP3000 via terminal lines. After repeated tests of the hardware, the problem seems to have disappeared. We decided not to replace any of the hardware until a hard failure occurs. Part replacement will not be a problem since there is an ample supply of spare boards at DARPA. Except for a delivery of software to DARPA, we do not expect any further major activity in this contract.

## 8 TCP FOR VAX UNIX

The major effort in the VAX TCP project this quarter was preparation of a public distribution of the TCP/IP software. This activity included work on fixing several long standing bugs in the networking code, adding some operational enhancements, and adding enhancements to the higher level protocol user software and utilities. In addition, some other kernel code enhancements were added in anticipation of future performance improvement work. The beta test sites were all surveyed for their reactions to the software, and to find out if there were any outstanding bugs that we should try to fix for the distribution.

### 8.1 Software Distribution

A general software release of the TCP/IP and related protocols was prepared, and announcement has been made of its availability. The distribution includes a complete set of 4.1BSD UNIX kernel sources, including the networking additions and modifications; user level programs that implement TELNET, FTP, and SMTP; utilities for network operation, status, and programming support; supporting libraries used in the higher level protocol programs, including a host name/address translation library; and documentation and installation instructions. The package is being made available to all sites

licensed for Berkeley 4.1BSD UNIX upon receipt of an executed supplemental licensing agreement and a \$300 duplication fee.

## 8.2 TCP/IP Work

### 8.2.1 Debugging

Several bugs in the TCP/IP code, some of long standing, were isolated and fixed in preparation for the public release. Several bugs caused infrequent data loss when doing large bulk transfers with FTP. This loss was traced to a race condition in buffer allocation, a bug in the buffer length adjustment routine, `m_adj`, and a bad sequence number comparison in the TCP packetization code.

Another bug resulted in large numbers of TCP resets being sent in response to incoming data for a non-existent connection, under certain conditions. In some cases, this bug could cause the entire system to deadlock. It was traced to a missing test in the TCP connection matching code which caused reset packets to be sent in response to incoming resets destined for non-existent connections. Also, we fixed a number of cases in the TCP packet preprocessor reset handling code that caused responses that were contrary to the TCP specification.

Finally, the UDP module which was recently added was fully debugged. The UDP is in use at MIT, with an implementation of TFTP, an alternative simplified file transfer protocol.

### 8.2.2 Enhancements

Much of the TCP/IP code was reorganized in anticipation of a future change to run the networking code as a software interrupt routine, rather than as a separate process. This involved merging several subroutines into common routines, and changing some of the subroutine call interfaces. In addition, the UCB data structure, which had been allocated from a fixed table, was changed to be allocated dynamically in network buffer space. This removes a restriction on the number of connections which can be open simultaneously.

A number of operational improvements were made. Ioctl calls (NETINIT and NETDISAB) were added to allow network interfaces to be enabled and disabled. Previously, the network subsystem was initialized at system boot time, and could not be disabled. Each interface may now be enabled or disabled individually. The NETINIT ioctl call assigns a network address to each interface and causes it to be initialized. This has simplified the network configuration procedure, since all that is needed to configure network interfaces is an entry in the system configuration file



for each network interface device. and an entry in a network configuration file, /etc/NETWORKS, that associates the network address with the interface's device name. A new utility, netinit, called from /etc/rc when the system goes into multi-user mode, reads /etc/NETWORKS, and issues the NETINIT ioctl calls. Gateway table initialization has also changed from a file read by the system at boot time, to NETGINIT ioctl calls which are issued from the mkgate utility. This utility is also called from /etc/rc and reads a standard gateway configuration file. /etc/GATEWAYS.

The TCP now sends and recognizes the maximum segment size option, which informs the connection peer of the maximum sized segment it is willing to receive. This allows TCPs which are willing to receive packets larger than the specified IP maximum (576 bytes) to do so, potentially increasing throughput. The maximum segment size option is now sent with each SYN packet, based on the maximum transmission unit for the network interface being used for the connection. On sending, the TCP uses this size to limit the size of the packets it sends. Previously, the limit was based exclusively on the ARPANET maximum transmission unit, which meant that packets sent outside the ARPANET could exceed the specified IP maximum.

Other enhancements which were added include options for non-blocking TCP connection opens, and blocking TCP connection closes. Previously (and now by default), TCP opens blocked until the connection was established. This presented a problem for applications like FTP (see below), in which two TCP connections must be coordinated by posting a listen on one, and notifying the peer of the existence of the listen on the other. Because of the blocking open, the listen and notification could not be correctly coordinated. A NETOWAIT ioctl was added to allow programs to explicitly block until connection establishment after non-blocking opens. The blocking close allows programs to wait until all TCP data structures are eliminated after a close is issued. Normally, close calls return immediately, even though the data structures persist until the closing protocol is completed.

Finally, the V2LNI ring network driver was modified to have an extended software defined protocol header that allows demultiplexing between higher level protocols, much like the ARPANET 1822 link number. Previously, there was no such demultiplexing information sent with each packet, so that only one higher level protocol (such as IP) could be sent over the ring. The new header format was defined by MIT.

### 8.3 Higher Level Protocol Support

The MTP mail protocol software was replaced by the newer SMTP protocol. While SMTP provides some new functionality, and removes some unused features from the older MTP protocol, the operational details of using SMTP remain the same. The SMTP was tested against other implementations at MIT, COMSAT, and DCEC.

Several bugs in the FTP user and server programs were fixed. Because of the blocking nature of the TCP open call, there was a synchronization problem in opening FTP data connections which often caused transfers to fail unnecessarily. Also, the definition of TCP FTP specifies that data connections on successive FTP transfers (within the same invocation of the program) use the same set of port numbers. This presents a problem, since the TCP closing protocol can take a long amount of time (up to 10 seconds, because of timers in the protocol), and the data structures persist during that time. This means that new connections occurring after the close might fail due to conflicts in port numbers. One solution to this problem is to delay between data connection closure and the next open. Another solution, which we chose, is to make the close call block until the previous connection's data structures disappear. This has the advantage of always preventing conflicts, and not always requiring long delays between transfers, depending on how the

closing sequence proceeded.

The new network host/address translation library was completed. This library is used by most of the higher level protocol and utility commands. It uses a flexible hashed data base of ASCII host names, addresses, and capabilities, that can be keyed on any of the fields. Several separate data bases can be merged into one host map, allowing groups of addresses from different sources, such as tables from the NIC and local network address tables. This library will be used in an internet name server implementation.

DISTRIBUTION

ARPA

Director (3 copies)  
Defense Advanced Research Projects Agency  
1400 Wilson Blvd.  
Arlington, VA 22209  
Attn: Program Manager  
R. Kahn  
V. Cerf  
R. Ohlander  
D. Adams

DEFENSE DOCUMENTATION CENTER (12 copies)

Cameron Station  
Alexandria, VA 22314

DEFENSE COMMUNICATIONS ENGINEERING CENTER

1860 Wiehle Road  
Reston, VA 22090  
Attn: Lt. Col. F. Zimmerman

DEPARTMENT OF DEFENSE

9800 Savage Road  
Ft. Meade, MD 20755  
R. McFarland C132 (2 copies)

DEFENSE COMMUNICATIONS AGENCY

8th and South Courthouse Road  
Arlington, VA 22204  
Attn: Code 252

NAVAL ELECTRONIC SYSTEMS COMMAND

Department of the Navy  
Washington, DC 20360  
B. Hughes, Code 6111  
F. Deckelman, Code 6131

BOLT BERANEK AND NEWMAN INC.

1701 North Fort Myer Drive  
Arlington, VA 22209  
E. Wolf

MIT Laboratory for Computer Science

545 Technology Square  
Cambridge, MA 02138  
D. Clark

DISTRIBUTION cont'd

BOLT BERANEK AND NEWMAN INC.

10 Moulton Street  
Cambridge, MA 02238

M. Brescia  
R. Bressler  
R. Brooks  
P. Cudhea  
W. Edmond  
L. Evenchik  
G. Falk  
J. Goodhue  
S. Groff  
R. Gurwitz  
J. Haverty  
F. Heart  
J. Herman  
R. Hinden  
D. Hunt  
E. Hunter  
S. Kent  
A. Lake  
W. Mann  
A. McKenzie  
D. McNeill  
W. Milliken  
A. Nemeth  
R. Rettberg  
R. Reynolds  
J. Robinson  
E. Rosen  
P. Santos  
J. Sax  
A. Sheltzer  
S. Storch  
R. Thomas  
B. Woznick  
Library

